Proiectarea filtrelor IIR prin metoda Prony

Laborator 3, PSS

Table of contents

1 Obiectiv

Proiectarea filtrelor IIR prin metoda Prony.

2 Noțiuni teoretice

2.1 Funcția de autocorelație

Pentru un semnal oarecare $x[n]$, funcția de autocorelație se definește ca:

$$r_{xx}[k] = \sum_{n=-\infty}^{\infty} x[n]x[n+k] \tag{1}$$

În Matlab, pentru un vector `x` de lungime `L` (elementele mergând de la `x[1]` la `x[L]`), funcția de autocorelație se calculează cu funcția `xcorr()`, ca în exemplul următor:

```
x = [1,2,3,4];
rxx = xcorr(x)    % Calculează autocorelația lui x
```

```
rxx =

    4.0000   11.0000   20.0000   30.0000   20.0000   11.0000    4.0000
```

În total sunt $2L-1$ valori (unde L = lungimea lui `x`), începând de la $r_{xx}[-(L-1)]$ și până la $r_{xx}[L-1]$. Așadar, valoarea $r_{xx}[0]$ din teorie se găsește de fapt în mijlocul vectorului rezultat, `rxx(L)`:

```
L = length(x);
rxx(L)      % Valoarea r_xx[0]
rxx(L+1)    % Valoarea r_xx[1]
rxx(L-3)    % Valoarea r_xx[-3]
```

2.2 Autocorelația parțială pentru metoda Prony

Pentru metoda Prony avem nevoie de valorile unei funcții de **autocorelație parțială**, definită ca:

$$r_{xx}[k,l] = r_{xx}[k-l] = \sum_{n=M+1}^{\infty} h[n-k]h[n-l] = \sum_{n=M+1-k}^{\infty} h[n]h[n+(k-l)] \qquad (2)$$

Diferența este că **suma nu începe de la** $n = 0$, ci de la o valoare superioară, astfel unele dintre primele elemente din sumă lipsesc.

Autocorelația parțială se poate calcula precum cea obișnuită, daca primele $M+1-max(k,l)$ elemente ale vectorului sunt transformate în 0. Autocorelația parțială $r_{xx}[k,l]$ este autocorelația în $(k-l)$ a vectorului astfel modificat.

Fie exemplul următor pentru a calcula $r_{xx}[k=1, l=2]$, cu $M = 2$:

```
M = 2;
x = [1,2,3,4];
k=1;
l=2;

x(1 : M+1-max(k,l)) = 0; % Setăm primele valori la 0
```

```
x                           % Afiseaza x modificat
rxx = xcorr(x)              % Calculeaza autocorelatia partiala, intreg vectorul

L = length(x);              % Ne intereseaza autocor. in punctul [k-l]
rxx(L+k-l)
```

2.3 Metoda Prony

În metoda Prony se calculează prima dată coeficienții $\{a_k\}$ dintr-un sistem de ecuații ce utilizează valorile autocorelației parțiale:

$$\begin{bmatrix} r_{dd}[1,1] & r_{dd}[1,2] & ... & r_{dd}[1,N] \\ r_{dd}[2,1] & r_{dd}[2,2] & ... & r_{dd}[2,N] \\ \vdots & ... & ... & \vdots \\ r_{dd}[N,1] & r_{dd}[N,2] & ... & r_{dd}[N,N] \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{bmatrix} = \begin{bmatrix} -r_{dd}[1,0] \\ -r_{dd}[2,0] \\ \vdots \\ -r_{dd}[N,0] \end{bmatrix} \tag{3}$$

Coeficienții b_k se obțin din aceleași ecuații ca la metoda Pade, înlocuind valorile $\{a_k\}$ găsite mai sus. Ecuațiile pentru b_k se pot scrie astfel:

$$b_n = h_d[n] + \sum_{k=1}^{N} a_k h_d[n-k]$$

sau, matricial:

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ ... \\ b_M \end{bmatrix} = \begin{bmatrix} h_d[0] \\ h_d[1] \\ h_d[2] \\ ... \\ h_d[M] \end{bmatrix} + \begin{bmatrix} 0 & 0 & \cdots & 0 \\ h_d[0] & 0 & \cdots & 0 \\ h_d[1] & h_d[0] & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ h_d[M-1] & h_d[M-2] & \cdots & h_d[M-N] \end{bmatrix} \cdot \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{bmatrix} \tag{4}$$

3 Exerciții teoretice

1. Folosiți metoda Prony pentru a afla parametrii sistemului cu următoarea funcție de sistem de ordinul 2:
$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

   care aproximează răspunsul la impuls dorit

$$h_d[n] = \{...0, \underset{\uparrow}{1}, 2, 3, 2, 1, 2, 3\}$$

   (originea timpului $n = 0$ este în dreptul primei valori de 1 din secvență).

4 Exerciții practice

1. Calculați și afișați funcția de autocorelație pentru vectorul constant $\{3,3,3,3,3,3,3\}$. Folosiți funcțiile `xcorr()` și `stem()`.

   Indicați apoi care sunt valorile lui $r_{xx}[0]$ și $r_{xx}[2]$

2. Creați o funcție `r = xcorr_prony(x, k, l, M)` pentru a calcula autocorelația parțială pentru a unui vector `x`. Funcția trebuie să returneze o singură valoare, $r_{xx}[k-l]$, pentru k și l specificați.

   Notă: nu uitați că $r_{xx}[0] =$ `rxx(L)` în Matlab.

   Testați funcția, verificând valorile urmatoare pentru `x = [1,2,3,2,1,2,3]` și `M=2`:

   - $r_{xx}[1,1] = 27$
   - $r_{xx}[1,2] = 22$
   - $r_{xx}[2,1] = 22$
   - $r_{xx}[2,2] = 31$
   - $r_{xx}[1,0] = r_{xx}[1] = 16$
   - $r_{xx}[2,0] = r_{xx}[2] = 14$

   Template:

```
function r = xcorr_prony(x, k, l, M)
% Computes restricted autocorrelation for the Prony method
% Inputs:
%   x = the input vector
%   k,l = the element to compute
%   M = the degree of the numerator polynomial B(z)
% Returns:
%   r = rxx[k-l]

...

end
```

3. Utilizați metoda Prony pentru a afla coeficienții a_k și b_k, pentru un sistem de ordinul 2 cu $M = 2$ și $N = 2$, și un răspuns la impuls dorit egal cu $h_d[n] = \{1,2,3,2,1,2,3\}$.

   Utilizați funcția `linsolve()` pentru a rezolva sistemul de ecuații al a_k.

```
hd = [1,2,3,2,1,2,3];
M = 2;              % numerator degree
N = 2;              % denominator degree
```

```
% Find coefficients a_k
A = ...   % 2x2 matrix
y = ...   % 2x1 column vector

a = linsolve(A,y)   % solve for a_k

b = [...] + [...]*a   % compute the b_k coefficients
% Find coefficients b_k
```

4. Implementați în Matlab o funcție generala pentru metoda Prony, pentru un sistem de orice ordin și orice semnal $h_d[n]$.

```
[b,a] = pronymet(ordin, hd)
```

Funcția va primi ca argumente:

- `ordin`: ordinul filtrului dorit
- `hd`: un vector cu răspunsul la impuls dorit

Funcția va returna coeficienții funcției de sistem a filtrului proiectat:

- `b`: coeficienții de la numărător
- `a`: coeficienții de la numitor

5. Folosiți metoda Prony pentru a găsi parametrii filtrului de ordin 2 care aproximează următoarea filtru de ordin superior (3):

$$H(z) = \frac{0.0736 + 0.0762z^{-1} + 0.0762z^{-1} + 0.0736z^{-3}}{1 - 1.3969z^{-1} + 0.8778z^{-1} - 0.1812z^{-3}}$$

a. Folosiți funcția `impz()` pentru a genera un răspuns la impuls suficient de lung al filtrului dat (de ex. 100 eșantioane);
b. Utilizați funcția `pronymet()` pentru a proiecta filtrul;
c. Reprezentați pe același grafic răspunsul la impuls al filtrului inițial și al celui proiectat, pentru primele 50 de eșantioane.

6. Să se încarce un semnal audio în Matlab și să se filtreze cu filtrul proiectat mai sus. Redați semnalul filtrat la ieșirea audio a sistemului.

5 Întrebări finale

1. TBD